Universidad Peruana de Ciencias Aplicadas

Curso: Fundamentos de Data Science

Sección: CC52

Profesor: Nérida Isabel Manrique Tunque

Entrega: Trabajo Parcial

Grupo N° 5

CÓDIGO	INTEGRANTE
U20191e742	Alcalde Gonzalez Renato Alberto
U202017033	Datto Aponte Antonio Francisco
U202124269	Diaz Villanueva Jeffrey Ulises

2023-01

1. CASO DE ANÁLISIS

El Dataset utilizado en el siguiente trabajo es acerca de datos de la demanda hotelera de dos hoteles de distinto tipo: uno es un Hotel Resort, y el otro es un Hotel Urbano. En este dataset hay un total de 119390 datos separados en 32 variables donde cada fila representa una reservación a uno de los hoteles. Este dataset fue hecho por Nuno Antonio, Ana de Almeida y Luis Nunes; los tres del Instituto Universitario de Lisboa (ISCTE-IUL) el 12 de diciembre de 2018.

El caso de uso que le estamos dando a esta información es para realizar un análisis sobre las reservas hechas en los distintos tipos de hoteles responder a ciertas preguntas acerca de la demanda de cada tipo de hotel teniendo en cuenta ciertos factores que pueden influir a la elección de los usuarios y en qué momentos es donde se dan los periodos de tiempo en los cuales se realizan la mayoría de las reservas. Esto puede ser beneficioso para diferentes hoteles para que sepan en qué momentos pueden llegar a tener más clientes y mejorar sus servicios para esos periodos de tiempo y que ofrecerle a los usuarios para que se decidan por hospedarse en su local.

2. CONJUNTO DE DATOS (DATA SET)

El dataset se basa en una cantidad total de 119390 con 32 variables distintas con información acerca de un hotel resort y uno urbano.

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_night
1	Resort Hotel	0	342	2015	July	27	1	
2	Resort Hotel	0	737	2015	July	27	1	
3	Resort Hotel	0	7	2015	July	27	1	
4	Resort Hotel	0	13	2015	July	27	1	
5	Resort Hotel	0	14	2015	July	27	1	
6	Resort Hotel	0	14	2015	July	27	1	
7	Resort Hotel	0	0	2015	July	27	1	
8	Resort Hotel	0	9	2015	July	27	1	
9	Resort Hotel	1	85	2015	July	27	1	
10	Resort Hotel	1	75	2015	July	27	1	
11	Resort Hotel	1	23	2015	July	27	1	
12	Resort Hotel	0	35	2015	July	27	1	
13	Resort Hotel	0	68	2015	July	27	1	
14	Resort Hotel	0	18	2015	July	27	1	
15	Resort Hotel	0	37	2015	July	27	1	
16	Resort Hotel	0	68	2015	July	27	1	
17	Resort Hotel	0	37	2015	July	27	1	
18	Resort Hotel	0	12	2015	July	27	1	
19	Resort Hotel	0	0	2015	July	27	1	

Cuenta con 17 columnas de tipo integer, 14 de tipo char y 1 de tipo num.

```
columnas_int <- sapply(hotel_bookings, is.integer)
cantidad_columnas_int <- sum(columnas_int)
```

```
cantidad_columnas_int    17L
```

```
columnas_char <- sapply(hotel_bookings, is.character)
cantidad_columnas_char <- sum(columnas_char)
```

cantidad_columnas_char	14L

Descripción de los datos:

- hotel: Representa el tipo de hotel: resort o urbano
- is _canceled: Representa si la reservación fue cancelada: 0 (no cancelada), 1 (cancelada)
- lead_time: Días que pasaron entre la reserva y el día de llegada del usuario
- arrival_date_year: Año de la llegada a la reservación
- arrival_date_month: Mes de la llegada a la reservación
- arrival_date_week_number: Número de semana de la llegada a la reservación
- arrival_date_day_of_month: Día del mes de la llegada a la reservación
- stays_in_weekend_nights: Número de noches del fin de semana que el usuario se quedó o reservó
- stay_in_week_nights: Número de noches de la semana que el usuario se quedó o reservó
- adults: Número de adultos que había en la reservación
- children: Número de niños que había en la reservación
- babies: Número de bebés que había en la reservación
- meal: Tipo de comida que se pidió para la reservación
- country: País de origen del hotel
- market_segmnet: Segmento del mercado
- distribution_channel: canal de distribución de reservas
- is_repeated_guest: Representa si el usuario ya ha reservado anteriormente: 0 (no), 1 (sí)
- previous_cancellations: Número de cancelaciones de reservas hechas por un usuario
- previous_booking_not_cancelled: Número de veces que el usuario no ha cancelado una reserva anteriormente
- reserved_room_type: Tipo de habitación reservada
- assigned_room_type: Tipo de habitación ofrecida al usuario
- booking_changes: Número de cambios en la reservación actual
- deposit_type: Tipo de depósito
- agent: ID de la agencia de viajes que hizo la reserva
- company: ID de la compañía responsable por el pago de la reservación
- days_in_waiting_list: Días en la lista de espera para las reservas
- costumer_type: Tipo de reserva
- adr: Tarifa diaria promedio
- required_car_parking_spaces: Aparcamientos requeridos en la reserva
- total_of_special_requests: Número de reservas especiales
- reservation_status: Estado de la reservación
- reservation_status_date: Dia del último estado de la reservación

3. ANÁLISIS EXPLORATORIO DE DATOS

a) Carga de datos:

En esta sección se carga con R el conjunto de datos "hotel_bookings.csv". Se leen los datos del archivo CSV y se establecen los parámetros correspondientes para la carga, como el

separador de columnas y la configuración de la línea de encabezado. Este paso es importante para poder trabajar con los datos.

```
#carga de datos
setwd("C:/Users/Propietario/Desktop/Clases/Ciclo 6")
hotel_bookings<-read.csv('hotel_bookings.csv', header=TRUE, sep=',',dec='.')
```

b) Inspeccionar datos:

En esta fase se realiza una comprobación inicial de los datos recién cargados. Utilizando la función “view”, el conjunto de datos se muestra en una tabla interactiva.

hotel_bookings	119390 obs. of 32 variables

La estructura del conjunto de datos se examina utilizando “str” para obtener una comprensión básica de la estructura y el contenido de los datos.

```
> str(hotel_bookings)
'data.frame':   119390 obs. of  32 variables:
 $ hotel                         : chr  "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...
 $ is_canceled                   : int  0 0 0 0 0 0 0 0 1 1 ...
 $ lead_time                     : int  342 737 7 13 14 14 0 9 85 75 ...
 $ arrival_date_year             : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ arrival_date_month            : chr  "July" "July" "July" "July" ...
 $ arrival_date_week_number      : int  27 27 27 27 27 27 27 27 27 27 ...
 $ arrival_date_day_of_month     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ stays_in_weekend_nights       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ stays_in_week_nights          : int  0 0 1 1 2 2 2 2 3 3 ...
 $ adults                        : int  2 2 1 1 2 2 2 2 2 2 ...
 $ children                      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ meal                          : chr  "BB" "BB" "BB" "BB" ...
 $ country                       : chr  "PRT" "PRT" "GBR" "GBR" ...
 $ market_segment                : chr  "Direct" "Direct" "Direct" "Corporate" ...
 $ distribution_channel          : chr  "Direct" "Direct" "Direct" "Corporate" ...
 $ is_repeated_guest             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_cancellations        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_bookings_not_canceled: int  0 0 0 0 0 0 0 0 0 0 ...
 $ reserved_room_type            : chr  "C" "C" "A" "A" ...
 $ assigned_room_type            : chr  "C" "C" "C" "A" ...
 $ booking_changes               : int  3 4 0 0 0 0 0 0 0 0 ...
 $ deposit_type                  : chr  "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...
 $ agent                         : chr  "NULL" "NULL" "NULL" "304" ...
 $ company                       : chr  "NULL" "NULL" "NULL" "NULL" ...
 $ days_in_waiting_list          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ customer_type                 : chr  "Transient" "Transient" "Transient" "Transient" ...
 $ adr                           : num  0 0 75 75 98 ...
 $ required_car_parking_spaces   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ total_of_special_requests     : int  0 0 0 0 1 1 0 1 1 0 ...
 $ reservation_status            : chr  "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...
 $ reservation_status_date       : chr  "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" ...
>
```

Los nombres de las columnas se comprueban utilizando “names”.

```
> names(hotel_bookings)
 [1] "hotel"                         "is_canceled"                   "lead_time"
 [4] "arrival_date_year"             "arrival_date_month"            "arrival_date_week_number"
 [7] "arrival_date_day_of_month"     "stays_in_weekend_nights"       "stays_in_week_nights"
[10] "adults"                        "children"                      "babies"
[13] "meal"                          "country"                       "market_segment"
[16] "distribution_channel"          "is_repeated_guest"             "previous_cancellations"
[19] "previous_bookings_not_canceled" "reserved_room_type"            "assigned_room_type"
[22] "booking_changes"               "deposit_type"                  "agent"
[25] "company"                       "days_in_waiting_list"          "customer_type"
[28] "adr"                           "required_car_parking_spaces"   "total_of_special_requests"
[31] "reservation_status"            "reservation_status_date"
>
```

c) Pre-procesar datos:

En esta fase, los datos se limpian para garantizar que estén listos para el análisis. Se identifican y procesan los valores que faltan en el conjunto de datos.

```
#viendo cuantos elementos vacios hay
sin_valor <- function(x){
  sum = 0
  for(i in 1:ncol(x))
  {
    cat("En la columna",colnames(x[i]),"total de valores NA:",colSums(is.na(x[i])),"\n")
  }
}
sin_valor(hotel_bookings)
```

```
> sin_valor(hotel_bookings)
En la columna hotel total de valores NA: 0
En la columna is_canceled total de valores NA: 0
En la columna lead_time total de valores NA: 0
En la columna arrival_date_year total de valores NA: 0
En la columna arrival_date_month total de valores NA: 0
En la columna arrival_date_week_number total de valores NA: 0
En la columna arrival_date_day_of_month total de valores NA: 0
En la columna stays_in_weekend_nights total de valores NA: 0
En la columna stays_in_week_nights total de valores NA: 0
En la columna adults total de valores NA: 0
En la columna children total de valores NA: 4
En la columna babies total de valores NA: 0
En la columna meal total de valores NA: 0
En la columna country total de valores NA: 0
En la columna market_segment total de valores NA: 0
En la columna distribution_channel total de valores NA: 0
En la columna is_repeated_guest total de valores NA: 0
En la columna previous_cancellations total de valores NA: 0
En la columna previous_bookings_not_canceled total de valores NA: 0
En la columna reserved_room_type total de valores NA: 0
En la columna assigned_room_type total de valores NA: 0
En la columna booking_changes total de valores NA: 0
En la columna deposit_type total de valores NA: 0
En la columna agent total de valores NA: 0
En la columna company total de valores NA: 0
En la columna days_in_waiting_list total de valores NA: 0
En la columna customer_type total de valores NA: 0
En la columna adr total de valores NA: 0
En la columna required_car_parking_spaces total de valores NA: 0
En la columna total_of_special_requests total de valores NA: 0
En la columna reservation_status total de valores NA: 0
En la columna reservation_status_date total de valores NA: 0
```

Después se aplica un proceso para rellenar los valores que faltan con valores apropiados; en este caso, se utiliza la media de la columna "children" para rellenar los valores vacíos apropiados.

```
#viendo la media de la columna children para rellenar los valores que estan en NA
summary(hotel_bookings$children)
```

```
> summary(hotel_bookings$children)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 0.0000  0.0000  0.0000  0.1039  0.0000 10.0000       4
```

```
#Rellenando los valores de NA con la media hallada anteriormente
hotel_bookings$children <- ifelse(is.na(hotel_bookings$children), 0, hotel_bookings$children)
```

Para los datos vacíos hemos optado por reemplazarlos utilizando la media y no eliminar dichas filas para que no haya ningún tipo de pérdida de datos que puedan afectar a las conclusiones.

A continuación, se utilizan dos métodos diferentes para tratar los valores atípicos. El primer método, denominado "Corrección método 1", utiliza los cuantiles para identificar los valores atípicos y sustituirlos por valores calculados, como la media y la mediana. El segundo método, denominado "Corrección método 2", utiliza el rango intercuartílico para identificar valores atípicos y sustituirlos por valores de corte específicos. La corrección de valores atípicos es esencial para garantizar que los valores atípicos no distorsionen los resultados de los análisis posteriores.

```r
#Outliers
install.packages("dplyr",dependencies = TRUE)
library(dplyr)
boxplot.stats(hotel_bookings$required_car_parking_spaces)
boxplot(hotel_bookings$required_car_parking_spaces)

# Correccion metodo 1
fix_outliers <- function(x, removeNA = TRUE){
  #Calculamos los quantiles 1) por arriba del 5% y por debajo del 95%
  9
  quantiles <- quantile(x, c(0.05, 0.95), na.rm = removeNA)
  x[x<quantiles[1]] <- mean(x, na.rm = removeNA)
  x[x>quantiles[2]] <- median(x, na.rm = removeNA)
  x
}

#correccion metodo 2
replace_outliers <- function(x, removeNA = TRUE){
  qrts <- quantile(x, probs = c(0.25, 0.75), na.rm = removeNA)
  # si el outlier esta por debajo del cuartil 0.5 o por arriba de 0.95
  caps <- quantile(x, probs = c(.05, .95), na.rm = removeNA)
  # Calculamos el rango intercuartilico
  iqr <- qrts[2]-qrts[1]
  # Calculamos el 1.5 veces el rango intercuartiligo (iqr)
  altura <- 1.5*iqr
  #reemplazamos del vector los outliers por debajo de 0.05 y 0.095
  10
  x[x<qrts[1]-altura] <- caps[1]
  x[x>qrts[2]+altura] <- caps[2]
  x
}
```

d) Visualizar datos:

En esta sección, se seleccionan las columnas de interés del conjunto de datos y se crea un nuevo conjunto de datos denominado "hotel_copy" que contiene sólo las variables relevantes para el desarrollo de las preguntas planteadas.

```r
#haciendo una copia del dataset manteniendo las cokumnas que consideramos necesarias
keeps<- c("hotel","arrival_date_month","arrival_date_year","children","babies","required_car_parking_spaces","is_canceled")
hotel_copy<-hotel_bookings[keeps]
head(hotel_copy)
```

Filter

	hotel	arrival_date_month	arrival_date_year	children	babies	required_car_parking_spaces	is_canceled
1	Resort Hotel	July	2015	0	0	0	0
2	Resort Hotel	July	2015	0	0	0	0
3	Resort Hotel	July	2015	0	0	0	0
4	Resort Hotel	July	2015	0	0	0	0
5	Resort Hotel	July	2015	0	0	0	0
6	Resort Hotel	July	2015	0	0	0	0
7	Resort Hotel	July	2015	0	0	0	0
8	Resort Hotel	July	2015	0	0	0	0
9	Resort Hotel	July	2015	0	0	0	1
10	Resort Hotel	July	2015	0	0	0	1
11	Resort Hotel	July	2015	0	0	0	1
12	Resort Hotel	July	2015	0	0	0	0
13	Resort Hotel	July	2015	0	0	0	0
14	Resort Hotel	July	2015	1	0	0	0
15	Resort Hotel	July	2015	0	0	0	0
16	Resort Hotel	July	2015	0	0	0	0
17	Resort Hotel	July	2015	0	0	0	0
18	Resort Hotel	July	2015	0	0	0	0
19	Resort Hotel	July	2015	0	0	0	0
20	Resort Hotel	July	2015	0	0	0	0
21	Resort Hotel	July	2015	0	0	0	0
22	Resort Hotel	July	2015	0	0	0	0

Además, este nuevo conjunto de datos se guarda como un archivo CSV denominado "hotel_copy.csv". La visualización y selección de variables es una parte importante de la preparación de los datos para su posterior análisis.

```r
#guardando el nuevo dataset como csv
write.csv(hotel_copy,"hotel_copy.csv",row.names=TRUE)
```

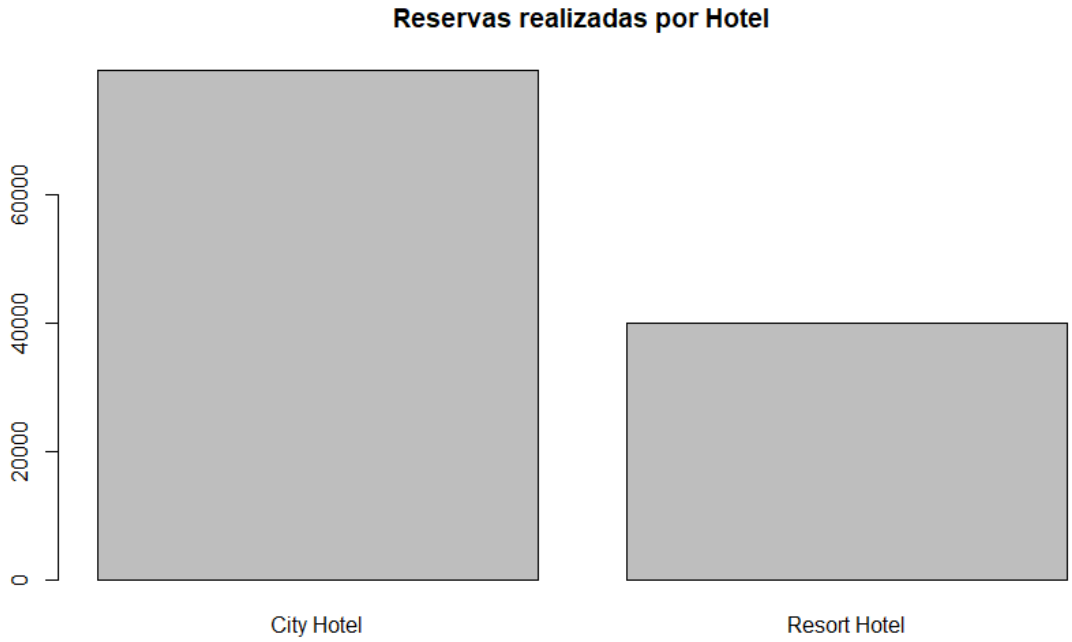
4. CONCLUSIONES PRELIMINARES

a. ¿Cuántas reservas se realizan por tipo de hotel? o ¿Qué tipo de hotel prefiere la gente?

```r
table(hotel_copy$hotel)
```

```r
barplot(table(hotel_copy$hotel), main="Reservas realizadas por Hotel",
        names= c("City Hotel", "Resort Hotel"))
```

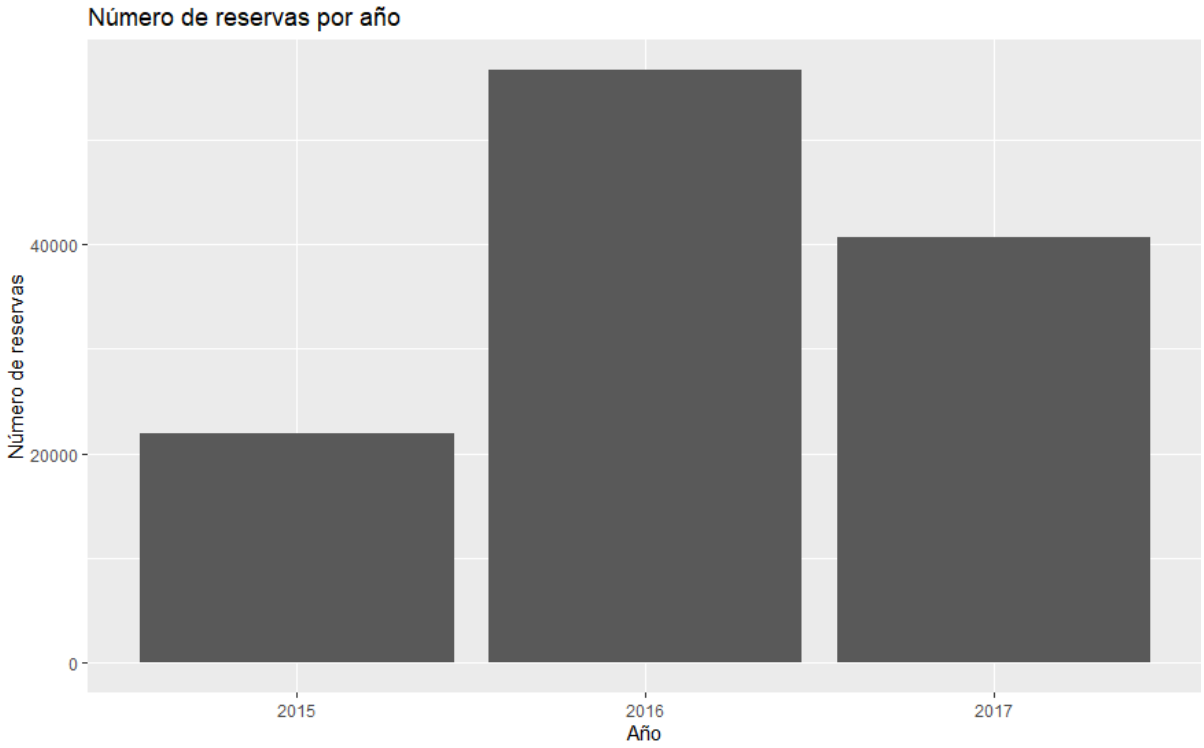
```
> table(hotel_copy$hotel)

  City Hotel Resort Hotel
       79330        40060
>
```



Utilizando la columna de “hotel” obtenemos la cantidad de reservas realizadas en ambos tipos de hoteles que hay. Según el resultado hay en total una cantidad de 79330 reservas hechas en “City Hotel” y 40060 en “Resort Hotel” por lo que podemos afirmar que el tipo de hotel que la gente prefiere hospedarse es en un “City Hotel”.

b. ¿Está aumentando la demanda con el tiempo?

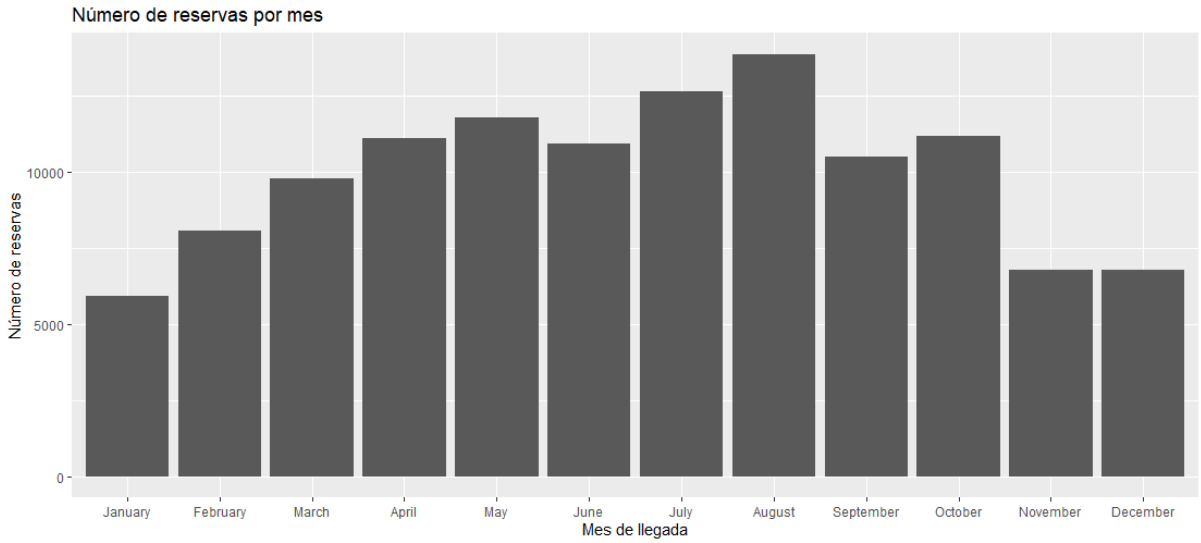
```
ggplot(hotel_copy, aes(x = factor(arrival_date_year))) +
  geom_bar() +
  labs(x = "Año", y = "Número de reservas") +
  ggtitle("Número de reservas por año")
```



Utilizando la columna de arrival_date_year podemos hacer un conteo de la cantidad de reservas totales que han habido en los tres distintos años registrados en el dataset.
Viendo el gráfico resultante podemos fijarnos que del 2015 al 2016 la demanda aumentó de manera significativa obteniendo más del doble de reservaciones que el año anterior. Sin embargo, del 2016 al 2017 vemos que la demanda disminuyó un poco si se compara con las reservaciones del primer año registrado. Teniendo esto en cuenta, podemos decir que la demanda está aumentando con el tiempo, con algunas ocasiones donde disminuyen ligeramente.

c. ¿Cuándo se producen las temporadas de reservas: alta, media y baja?

```
ggplot(hotel_copy, aes(x = factor(hotel_copy$arrival_date_month, levels = month.name))) +
  geom_bar() +
  labs(x = "Mes de llegada", y = "Número de reservas") +
  ggtitle("Número de reservas por mes")
```



En base a la columna arrival_date_month, se ha generado el gráfico 'Número de reservas por mes'. Se puede observar que durante los meses de Agosto y Julio se han presentado mayor número de reservas de habitaciones respecto al resto de meses. También se ve que Noviembre, Diciembre y Enero son meses en donde no suele haber muchas reservas en comparación al resto del año.
Entonces, concluimos que:

- La temporada de alta demanda de reservas es durante los meses Julio y Agosto.
- La temporada de baja demanda de reservas es durante los meses de Noviembre, Diciembre y Enero.
- La temporada de media demanda de reservas es durante los meses de Marzo, Abril, Mayo, Junio, Septiembre y Octubre.

d. ¿Cuándo es menor la demanda de reservas?

```
table(hotel_copy$arrival_date_month)
```

```
> table(hotel_copy$arrival_date_month)

    April    August  December  February   January      July
    11089     13877      6780      8068      5929     12661
     June     March       May  November   October September
    10939      9794     11791      6794     11160     10508
```

Según el gráfico de 'Número de reservas por mes' en la resolución de la pregunta c, se observa que la menor demanda de reservas ocurre en los meses de Noviembre, Diciembre y Enero. Siendo Enero el menor de los tres, con 5929 reservas.

e. ¿Cuántas reservas incluyen niños y/o bebes?

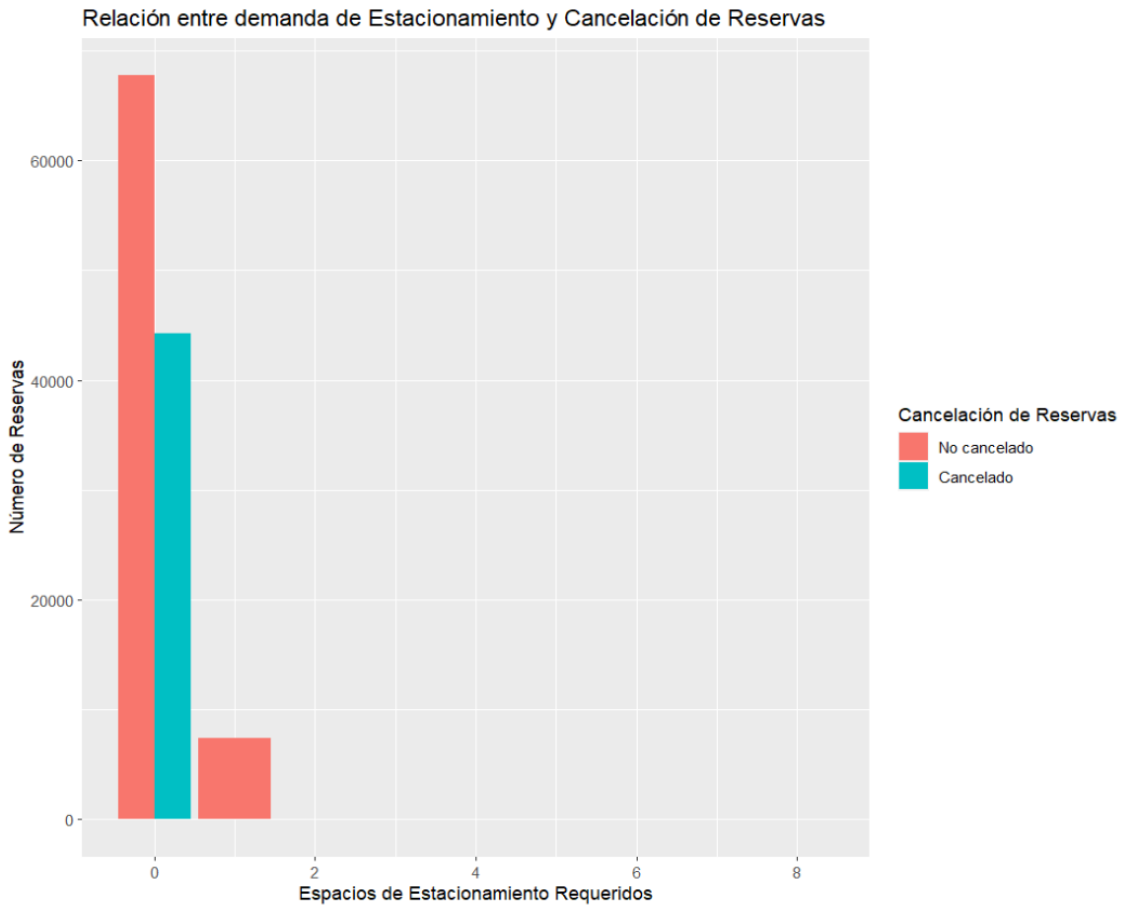
```
reservas_con_ninos_o_bebes <- subset(hotel_copy, children > 0 | babies > 0)
cantidad_reservas_con_ninos_o_bebes <- nrow(reservas_con_ninos_o_bebes)
cat("El número de reservas que incluyen niños y/o bebés es:", cantidad_reservas_con_ninos_o_bebes, "\n")
```

```
>
> reservas_con_ninos_o_bebes <- subset(datos, children > 0 | babies > 0)
> cantidad_reservas_con_ninos_o_bebes <- nrow(reservas_con_ninos_o_bebes)
> cat("El número de reservas que incluyen niños y/o bebés es:", cantidad_reservas_con_ninos_o_bebes, "\n")
El número de reservas que incluyen niños y/o bebés es: 9332
>
```

En base a las columnas babies, children se contabiliza el número de reservas con niños o bebes. Se observa en la ejecución del código que la contabilización de todas las reservas que contaban con al menos un niño o bebe fue de 9332.
Entonces existen 9332 reservas que incluyen niños y/o bebes.

f. ¿Es importante contar con espacios de estacionamiento?
Para determinar si los espacios de estacionamiento son importantes, podemos analizar la relación entre la demanda de espacios de estacionamiento y la disponibilidad de las mismas. Creemos primero un gráfico de barras que muestre el número de reservas en función de la disponibilidad de plazas de aparcamiento.

```
# Crear un gráfico de barras que muestra la cantidad de reservas según la disponibilidad de estacionamiento
ggplot(hotel_copy, aes(x = required_car_parking_spaces, fill = as.factor(is_canceled))) +
  geom_bar(position = "dodge") +
  labs(title = "Relación entre demanda de Estacionamiento y Cancelación de Reservas",
       x = "Espacios de Estacionamiento Requeridos",
       y = "Número de Reservas",
       fill = "Cancelación de Reservas") +
  scale_fill_discrete(name = "Cancelación de Reservas", labels = c("No cancelado", "Cancelado"))
```



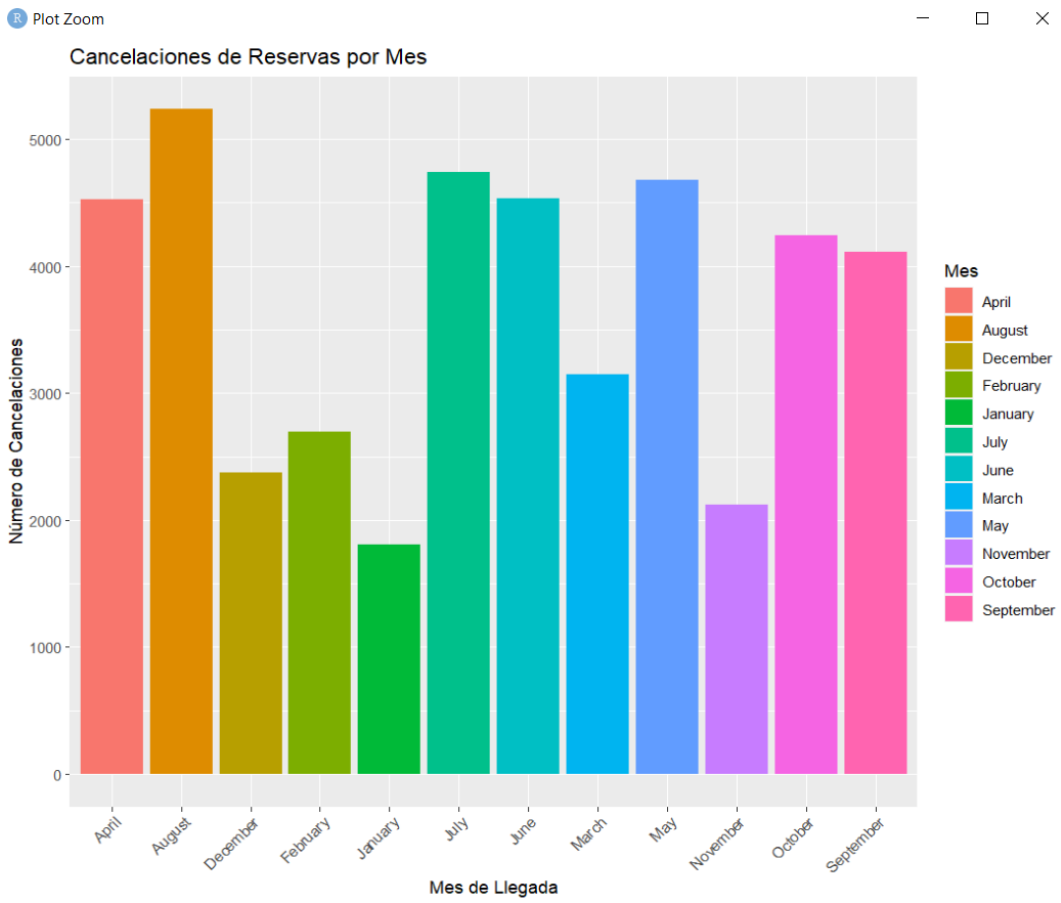
De los datos se desprende que el porcentaje de reservas canceladas es del 39,5% para las reservas que no requieren aparcamiento (required_car_parking_spaces = 0). Aunque

algunas de estas reservas fueron canceladas, la mayoría, alrededor del 60,5%, no lo fueron. Esto sugiere que la disponibilidad de plazas de aparcamiento puede influir en la decisión de cancelar una reserva, pero no es el único factor determinante. Otros factores también pueden desempeñar un papel importante en la decisión de cancelar una reserva.

g. ¿En qué meses del año se producen más cancelaciones de reservas?

Para identificar los meses con más cancelaciones, podemos crear un gráfico de barras que muestre el número de cancelaciones por mes.

```
# Crear un gráfico de barras que muestra la cantidad de cancelaciones por mes
cancelaciones_por_mes <- hotel_copy %>%
  filter(is_canceled == 1) %>%
  group_by(arrival_date_month) %>%
  summarise(total_cancelaciones = n())

ggplot(cancelaciones_por_mes, aes(x = arrival_date_month, y = total_cancelaciones, fill = arrival_date_month)) +
  geom_bar(stat = "identity") +
  labs(title = "Cancelaciones de Reservas por Mes",
       x = "Mes de Llegada",
       y = "Número de Cancelaciones") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  scale_fill_discrete(name = "Mes")
```



Según los datos, los meses con mayor número de cancelaciones, en orden descendente, son los siguientes

- Agosto (5239 cancelaciones)
- Julio (4.742 anulaciones)
- Mayo (4.677 anulaciones)

- Junio (4.535 anulaciones)
- Abril (4.524 anulaciones)

En estos meses, el número de cancelaciones es superior al de los demás meses del año. Esto puede estar relacionado con factores estacionales, días festivos o eventos que afectan a la demanda de reservas en estos meses.

5. LINK DEL GITHUB:
https://github.com/J3ffo3/CC216-TP-2023-2-CC52-Grupo5